Pontifícia Universidade Católica
do Rio de Janeiro

Luiz Felipe de Souza e Silva

Inspeção Não-Intrusiva da Comunicação em Aplicações Baseadas em RPC

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática da PUC-Rio

Orientador: Prof. Renato Fontoura de Gusmão Cerqueira

Rio de Janeiro
Abril de 2013

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Luiz Felipe de Souza e Silva

Inspeção Não-Intrusiva da Comunicação em Aplicações Baseadas em RPC

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Renato Fontoura de Gusmão Cerqueira
Orientador
Departamento de Informática – PUC-Rio

Prof. Marcus Endler
Departamento de Informática – PUC-Rio

Prof. Alexandre Sztajnberg
Universidade do Estado do Rio de Janeiro

Prof. José Eugênio Leal
Coordenador Setorial do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 8 de abril de 2013

Luiz Felipe de Souza e Silva

Graduou-se em Engenharia Elétrica com ênfase em Eletrônica e Computação na UFRJ – Universidade Federal do Rio de Janeiro – em 2007. Trabalhou na Globo.com como engenheiro de webmedia até Junho de 2008. Certificado RHCE em 2010, atualmente trabalha como analista de sistemas com ênfase em infraestrutura na gerência de Tecnologia Geofísica da Petrobras S/A.

Ficha Catalográfica

Souza e Silva, Luiz Felipe de

Inspeção Não-Intrusiva da Comunicação em Aplicações Baseadas em RPC / Luiz Felipe de Souza e Silva ; orientador: Renato Fontoura de Gusmão Cerqueira. – Rio de Janeiro : PUC-Rio, Departamento de Informática, 2013.

[], 85 f. : il. (color.) ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2013.

Inclui bibliografia

1. Informática – Teses. 2. Depuração de aplicações distribuídas. 3. Chamadas de sistemas. 4. Monitoramento não intrusivo. 5. Interceptação de chamadas. I. Cerqueira, Renato Fontoura de Gusmão. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

Grandes projetos nunca são concluídos pela mão de um único indivíduo. Mesmo quando, devido a uma determinada conquista, apenas uma pessoa fica em evidência, esta não a conquistou sem recorrer à ajuda de terceiros, sejam elas entidades religiosas, orientadores acadêmicos, pais, família, amigos ou colegas de trabalho. Convergente a este raciocínio, este projeto não foi diferente. Devo muito a Deus, a minha família, e as demais pessoas que me ajudaram ao longo deste longo caminho por todo apoio nos momentos difíceis e a compreensão pelas ausências ao longo desses últimos anos, o que creio que seja comum a todos os mestrandos.

Em especial, agradeço, na Petrobras, ao Luis Antônio, meu coordenador, e ao Edson Yoshino, meu gerente imediato, por terem buscado, na empresa, o patrocínio necessário para que eu pudesse concluir o mestrado, mesmo quando não havia condição favorável para tal. Na PUC-Rio, não posso deixar de agradecer ao meu orientador acadêmico, professor Renato Cerqueira, por todo apoio, empenho e dedicação para que houvesse sucesso no meu ingresso no programa de pós-graduação desta entidade, e também por todo período que frequentei o programa, sem esquecer da professora Noemi Rodriguez, por sua enorme paciência em me escutar e aconselhar nos momentos mais tensos.

Aos amigos, Roberto Gonzalez, por ter dado a semente que culminou neste trabalho, e ao Amadeu Barbosa, pelo seu apoio incondicional nos instantes finais e decisivos para conclusão deste trabalho.

Resumo

Souza e Silva, Luiz Felipe de; Cerqueira, Renato Fontoura de Gusmão (Orientador). **Inspeção Não-Intrusiva da Comunicação em Aplicações Baseadas em RPC.** Rio de Janeiro, 2013. 85p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A depuração de software é uma atividade que tipicamente demanda um grande esforço, fundamentalmente devido à necessidade de analisar múltiplas condições que determinam o contexto de execução do software e de criar suposições que expliquem o motivo do defeito, para, só assim, sermos capazes de corrigi-lo. No caso de sistemas distribuídos, o paralelismo, a ordem de escalonamento, os atrasos na comunicação e falhas nos equipamentos são exemplos de fatores que aumentam ainda mais a complexidade da atividade de depuração. Portanto, a busca por ferramentas que auxiliem neste processo é contínua. Neste trabalho, propomos uma ferramenta de monitoração e visualização, não intrusiva, da comunicação entre componentes de sistemas distribuídos, através do uso de analisadores de protocolo de comunicação e da monitoração das chamadas de sistema de leitura e gravação.

Palavras-chave

Depuração de aplicações distribuídas; chamadas de sistema; monitoramento não intrusivo; interceptação de chamadas.

Abstract

Souza e Silva, Luiz Felipe de; Cerqueira, Renato Fontoura de Gusmão (Advisor). **Non-Intrusive Communication Inspection in RPC-Based Applications.** Rio de Janeiro, 2013. 85p. MSc. Dissertation – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Software debugging is an activity that typically requires an huge effort, primarily due to the need to analyze multiple conditions that determine the execution context of the software and to create assumptions that explain the problem, for only thus being able to fix it. In the case of distributed systems, parallelism, order scheduling, delays in communication and equipment failures are examples of factors that further increase the complexity of the debugging activity. Therefore, the search for tools that assist in this process is continuous. In this dissertation, we propose a tool for non-intrusive monitoring and visualization of communications between components of distributed systems, based on communication protocol analyzers and monitoring of read and write system calls.

Keywords

Debugging of Distributed Systems; System Calls; Non-intrusive Monitoring; Call Interception.

Sumário

1 Introdução 12
1.1 Ciclo de depuração de um software 13
1.2 Depuração aplicada a sistemas distribuídos 13
1.3 Objetivos 14
1.4 Contribuições 16
1.5 Estrutura do documento 16

2 Metodologias, técnicas e soluções de depuração 17
2.1 Metodologias 17
2.1.1 Top-down 18
2.1.2 Bottom-up 18
2.1.3 Depuração em duas fases 18
2.2 Técnicas 19
2.2.1 Impressão de mensagens 19
2.2.2 Rastreamento 20
2.2.3 Pontos de parada 21
2.2.4 Execução assertiva 21
2.2.5 Execução controlada 22
2.2.6 Monitoração 22
2.2.7 Execução reprodutiva 22
2.3 Soluções 23
2.3.1 Soluções em hardware 24
2.3.2 Soluções no núcleo do sistema operacional 24
2.3.3 Soluções por bibliotecas 24
2.4 Considerações finais 25

3 A ferramenta proposta 26
3.1 Modelagem da ferramenta 27
3.2 Instrumentação do núcleo do SO 29
3.2.1 Monitoramento 32
3.2.2 Dispositivo de comunicação com o modo usuário 35
3.2.3 Interface de gerência 37
3.3 Formato das mensagens de instrumentação 38
3.4 GUI com o usuário 43
3.5 Metodologias de utilização 47

4 Avaliações e resultados 51
4.1 Desempenho do código inserido no núcleo 51
4.1.1 Degradação no ponto de entrada 54
4.1.2 Degradação ao gravar no buffer circular 61
4.2 Avaliação do desempenho da interface núcleo-usuário 68
4.3 Avaliação do desempenho da interface GUI 69
4.4 Avaliação funcional 70

4.4.1 Preparação do ambiente para testes 71
4.4.2 Conexão a um servidor de mensagens 72
4.4.3 Troca de mensagens CORBA utilizando DII/DSI 73
4.4.4 Gerenciador de pregão eletrônico 74
4.4.5 Gerenciador de pregão eletrônico com notificações 76
4.5 Considerações finais 78

5 Conclusão 80
5.1 Trabalhos futuros 81

6 Referências Bibliográficas 82

Lista de Figuras

Figura 2.1 Rastreamento da biblioteca C padrão 20
Figura 2.2 Rastreamento de chamadas a bibliotecas 20
Figura 2.3 Rastreamento de chamadas de sistema 21
Figura 2.4 Dr. Watson para Windows NT 21
Figura 2.5 Falha de assertividade do runtime do Visual C 22
Figura 3.1 Representação gráfica da ferramenta 26
Figura 3.2 Indireções formadas pelo vetor de chamadas de sistema 27
Figura 3.3 Executando uma chamada de sistema 27
Figura 3.4 Separação de privilégios nos processadores x86 28
Figura 3.5 Entrada/saída da monitoração de leitura 30
Figura 3.6 O módulo, antes e depois 31
Figura 3.7 Isolamento do monitoramento e interface núcleo-usuário 32
Figura 3.8 Atribuições do registrador CR0 33
Figura 3.9 Pseudocódigo do método da força bruta 33
Figura 3.10 Os dois vetores de chamadas de sistema 34
Figura 3.11 O algoritmo de interceptação 35
Figura 3.12 Mensagem de saída apresentada após a carga do módulo 36
Figura 3.13 Status do funcionamento do módulo 37
Figura 3.14 Comando para monitorar um processo 37
Figura 3.15 Comando para remover monitoração de um processo 38
Figura 3.16 Incluindo processos parentes na monitoração 38
Figura 3.17 Cabeçalho utilizado pelo formato *libpcap* 39
Figura 3.18 O processo de encapsulamento de um pacote TCP/IP 40
Figura 3.19 Binary Encoding Representation 42
Figura 3.20 A interface do *wireshark* 43
Figura 3.21 Campo opcional IP com as informações de processo 44
Figura 3.22 Campo opcional IP manipulado pela extensão 44
Figura 3.23 Acesso ao filtro rápido do Wireshark 45
Figura 3.24 Resultado da aplicação do filtro 45
Figura 3.25 Interpretação nativa do protocolo GIOP 46
Figura 3.26 Uso da IDL para obter mais informações do pacote 46
Figura 3.27 Acesso ao filtro rápido no Wireshark 47
Figura 3.28 Elos de comunicação entre componentes A e B 48
Figura 3.29 Topologias de comunicação 49
Figura 4.1 Assinatura das chamadas de sistema *read*() e *write*() 52
Figura 4.2 Cálculo da quantidade de eventos necessários para ler um arquivo 52
Figura 4.3 Diagrama do algoritmo aplicado na função de entrada 53
Figura 4.4 Diagrama do algoritmo de identificação do *socket* 54
Figura 4.5 Desempenho de E/S sem carga extra 55
Figura 4.6 Desempenho de E/S com a monitoração habilitada 56
Figura 4.7 Desempenho de E/S do processo monitorado 57
Figura 4.8 Utilização de CPU com os testes da seção 4.1.1.2 59
Figura 4.9 Processos concorrentes observados nos testes da seção 4.1.1.2 59

Figura 4.10 Utilização de CPU com os testes da seção 4.1.1.3 60
Figura 4.11 Concorrência dos processos nos testes da seção 4.1.1.3 61
Figura 4.12 Desempenho de E/S de rede sem carga extra 63
Figura 4.13 Desempenho de E/S de rede com a monitoração habilitada 64
Figura 4.14 Desempenho de E/S de rede do processo monitorado 65
Figura 4.15 Utilização de CPU com o teste da seção 4.1.2.2 66
Figura 4.16 Utilização de CPU com o teste da seção 4.1.2.3 66
Figura 4.17 Concorrência dos processos do teste da seção 4.1.2.2 67
Figura 4.18 Concorrência dos processos do teste da seção 4.1.2.3 67
Figura 4.19 Desempenho da transformação núcleo → usuário 69
Figura 4.20 Desempenho de carga de registros do *Wireshark* 70
Figura 4.21 Carregamento do módulo de interceptação de chamadas de sistema . 71
Figura 4.22 Criação do vínculo de comunicação núcleo-usuário 71
Figura 4.23 Instanciação da camada núcleo-usuário 71
Figura 4.24 Instanciação da interface GUI 72
Figura 4.25 Captura utilizando o *Wireshark* 72
Figura 4.26 Captura utilizando a ferramenta 72
Figura 4.27 Stream do sniffer 73
Figura 4.28 Stream da ferramenta 73
Figura 4.29 Folha obtida com a carga da IDL forjada 73
Figura 4.30 Pesquisa utilizando características do pacote GIOP 74
Figura 4.31 Diagrama do software de gerenciamento de pregões 74
Figura 4.32 Resultado da monitoração do *Auction Server* 75
Figura 4.33 Conteúdo da exceção de sistema encapsulada no pacote GIOP 75
Figura 4.34 Interface do software de pregão eletrônico com notificações 76
Figura 4.35 Monitoração do software de pregão eletrônico com notificações 77
Figura 4.36 Aplicação de filtros para remoção das entradas TCP 78

Lista de Tabelas

Tabela 3.1 Estrutura utilizada para a comunicação núcleo-usuário 37
Tabela 3.2 Cabeçalho global utilizado pela *libpcap* 39
Tabela 3.3 Cabeçalho de cada pacote armazenado no arquivo *libpcap* 39
Tabela 3.4 Estrutura de um pacote TCP/IP 41
Tabela 3.5 Estrutura de um cabeçalho IP opcional 41
Tabela 3.6 Campo opcional IP com dados de processo e descritor 42
Tabela 3.7 Mensagens monitoradas pelo componente A 48
Tabela 3.8 Mensagens monitoradas pelo componente B 48
Tabela 4.1 Médias de desempenho obtidas sem carga extra 56
Tabela 4.2 Médias de desempenho com a monitoração habilitada 57
Tabela 4.3 Médias de desempenho do processo monitorado 58
Tabela 4.4 Incremento de tempo de E/S percentual da seção 4.1.1.2 / 4.1.1.1 58
Tabela 4.5 Incremento de tempo de E/S percentual da seção 4.1.1.3 / 4.1.1.1 60
Tabela 4.6 Diferença percentual relativa entre as tabelas 4.5 e 4.4 60
Tabela 4.7 Médias de desempenho de rede obtidas sem carga extra 62
Tabela 4.8 Médias de desempenho de rede obtidas com a monitoração habilitada 63
Tabela 4.9 Médias de desempenho do processo monitorado 64
Tabela 4.10 Incremento de tempo de E/S percentual da seção 4.1.2.2 / 4.1.2.1 ... 65
Tabela 4.11 Incremento de tempo de E/S percentual da seção 4.1.2.3 / 4.1.2.1 ... 65
Tabela 4.12 Diferença percentual relativa entre as tabelas 4.9 e 4.8 68
Tabela 4.13 Médias de desempenho da interface 68
Tabela 4.14 Médias de desempenho do *Wireshark* 69